



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/690,918	10/21/2003	Michael M. Klock	NVID-061/00US	6921
23419	7590	12/01/2006		EXAMINER
COOLEY GODWARD KRONISH LLP 3000 EL CAMINO REAL 5 PALO ALTO SQUARE PALO ALTO, CA 94306			WOODS, ERIC V	
			ART UNIT	PAPER NUMBER
			2628	

DATE MAILED: 12/01/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	10/690,918	KLOCK ET AL.	
Examiner	<b>Art Unit</b>		
Eric Woods	2628		

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

1)  Responsive to communication(s) filed on 23 October 2006.  
2a)  This action is **FINAL**.                            2b)  This action is non-final.  
3)  Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

4)  Claim(s) 1-6,8 and 10-32 is/are pending in the application.  
4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.

5)  Claim(s) \_\_\_\_\_ is/are allowed.

6)  Claim(s) 1-6,8 and 10-32 is/are rejected.

7)  Claim(s) \_\_\_\_\_ is/are objected to.

8)  Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

9)  The specification is objected to by the Examiner.

10)  The drawing(s) filed on \_\_\_\_\_ is/are: a)  accepted or b)  objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)  The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)  Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a)  All b)  Some \* c)  None of:  
1.  Certified copies of the priority documents have been received.  
2.  Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
3.  Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1)  Notice of References Cited (PTO-892)  
2)  Notice of Draftsperson's Patent Drawing Review (PTO-948)  
3)  Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_.

4)  Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_.  
5)  Notice of Informal Patent Application  
6)  Other: \_\_\_\_.

## DETAILED ACTION

### ***Response to Arguments***

Applicant's arguments, see Remarks pages 1-3 and claim amendments, filed 10/23/2006, with respect to the rejection(s) of claim(s) 1-6, 8, and 10-32 under various statutes have been fully considered and are persuasive.

Therefore, in light of applicant's amendments to the claims, the rejection of claim 28 under 35 USC 101 has been withdrawn.

In view of applicant's amendments to the claims, the rejection of claims 1-6, 8, and 10-32 under 35 USC 103(a) stand withdrawn.

However, upon further consideration, a new ground(s) of rejection is made in view of various references as set forth below.

The recitation of the term 'automatically' to the retrieval of data is only automation of a task previously done manually, e.g. instead of the user manually selecting the file the computer selects it instead. This is clearly merely automating a task previously done manually, and thusly is rejected as an obvious expedient as per *In re Venner*, 262 F.2d 91, 95, 120 USPQ 193, 194 (CCPA 1958), where the court held that broadly providing an automatic or mechanical means to replace a manual activity which accomplished the same result is not sufficient to distinguish over the prior art, where the 'automated' retrieval of data would be no different than the timer means introduced to control the release of the engine piston, because in this case the file is being retrieved as part of a human activity, e.g. the use of the input device, which proves clearly that the automation is still triggered by a human being, clearly meeting the criteria under

*Venner.*

A closer reading of the relevant case law, namely *In re Venner* shows that simply because a computer (or broadly, "automated means", which in the case of *Venner* happened to be a timer) is used to perform a step previously performed by a human being (in *Venner*, the step was determining when to release the relevant engine part from the mold) does not make it patentable or non-obvious (see MPEP 2106 and specifically 2144.04, section (III)). Further, the obviousness rejection in that case was upheld at least partially because the user of the system still had to choose the point at which the timer was initiated, so even though automatic means were used to release the mold, the user still had to initiate the process. Therefore, on both grounds - both broadly that automatically positioning views as applicant recites is merely automating an activity previously manually done by a user is *per se* only automating a previously manual activity, and that specifically in respect to *Venner*, that the present step is still **initiated by the user at a time of the user's choosing, and the user chooses which parameters will be optimized, and (although the claim does not specifically say so) the user (as is well known in the PC art) can / could choose the parameters to be optimized and their ranges as applied to the graphics card in question.** As such, the activity is still manual in nature, with only a small step converted to an automatic action by a computer, as applicant clearly admits on page 1 of the Remarks states that a user operating a general-purpose computer could in fact perform all the steps, but then asserts that having the user set up and test a matrix of overclocking parameters has several problems.

However, to sustain a holding of obviousness it is only necessary that the automation occur, **not** whether or not the automation of several tedious tasks, irrespective of the merits of such automation, unless there is either a) special circumstance or b) some new invention in the automation itself. Applicant has submitted no documentation in the specification or otherwise that would justify a holding of special circumstances (e.g. long-felt need, commercial success, and/or unexpected results).

Therefore, the conclusion that one of ordinary skill in the computer art, which in this case would be assumed to be someone of at least a bachelor's degree in computer engineering or science with a focus on computer graphics (justified because it is reasonable to hold that the minimum educational background to become a patent examiner in an art area would be necessary to be qualified as 'one of ordinary skill') would find it expedient to write a program or the like perform those steps (e.g. to automate them) is justified. Note that both the Cooper and Feierbach references teach automation of such testing (e.g. an automatic test).

It is further noted that the applicant did not challenge the taking of Official Notice with respect to claims 8 and 19, thusly barring any challenges on appeal. Thusly, those takings of Official Notice have been entered into the record as facts assumed to be in evidence with respect to any further proceedings in this matter.

It is further noted that applicant's own software (e.g. CoolBits) was available over a year before the filing date of the instant application and allowed overclocking of NVidia graphics cards.

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 1, 3-4, 6, 10-17, 20, and 22-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bigjakkstaffa (<http://www.sysopt.com/articles/VCOGuide/>) in view of Feierbach (US 2001/0009022 A1) and Cooper et al (US PGPub 2002/0143488 A).

As to claim 1,

Bigjakkstaffa teaches the following limitations:

A computer implemented method of overclocking a graphics system, comprising:  
(Bigjakkstaffa page 3 shows a screen where the user can tune the core and memory frequencies, e.g. speeds)

-Receiving a user request for overclocking; (Bigjakkstaffa, the overclocking menu tab on page 3, and see page 5, first paragraph, wherein the user requests overclocking by

the window. Bigjakkstaffa clearly teaches overclocking of at least one of a graphics processor and a graphics memory (see Bigjakkstaffa, pages 5 and 6, wherein the overclocking parameters to be overclocked are the clock of the GPU core and the clock of the GPU memory))

-Forming sets of overclocking parameters to be evaluated from a set of supported overclocking parameters pre-selected for said graphics system that includes a set of graphics processor core clock rates and memory clock rates, having an initial starting point and a maximum end point associated with a graphics processor and a graphics memory; (Bigjakkstaffa, pages 5 and 6, wherein the overclocking parameters to be evaluated are the clock of the GPU core and the clock of the GPU memory, where the user can clearly vary the setting of each; indeed, Bigjakkstaffa teaches on page 3 that each parameter can be varied independently, whilst teaching on page 6 that both can be varied simultaneously and/or independently, note that the **stock** referred speeds on page 5 have are 250MHz/513MHz, which have a ratio of 2.052:1, where Bigjakkstaffa then recommends changing them in increments of 10MHz, which would change the ratio between them accordingly. Next, the idea of varying them individually is discussed by Bigjakkstaffa in page 8, where a final overclocked graphics card is shown as having speeds of 280MHz/600MHz, which has a ratio of 2.143:1 (page 9), where the overall memory speed is then suggested to be lowered 20MHz (by itself). Therefore, for at least all the above reasons, both memory speed and graphics core speed can/may be adjusted independently and such adjustment is taught and/or suggested. These parameters are unique to the components in question. Now, the concept of different

sets of parameters can be represented as different groupings of parameters tried by the user (e.g. 10MHz stepped intervals suggested above). Finally, the windows shown in Bigjakkstaffa **clearly** indicate that the set of overclocking parameters has a known upper and a known lower limit, as in on page 3 core clock can be adjusted between 125MHz and 375MHz and the like with respect to the memory clock. This would clearly define sets of graphics core clock rates and memory clocks rates that have an “initial starting point and a maximum end point”, where these are clearly associated with a graphics processor and a graphics memory)

-For each set of overclocking parameters, automatically applying a stress test, said stress test executing a graphics test sequence and monitoring pixel errors of said graphics system; and (Bigjakkstaffa, pages 6 and 7, wherein the user clicks on the “test” button, and wherein further tests are performed by running a 3dmark bench, and if no visual artifacts or texture tearing appears, then the user defined GPU core and GPU clock speeds pass the test and are determined to be a safe set of overclocking parameters. Bigjakkstaffa inherently performs graphics test sequences (Firstly, a Benchmark, such as the 3Dmark2001 and the like, are graphical test sequences) and monitors texture tearing and visual artifacts, except that the user is doing so. The rationale for modifying Bigjakkstaffa to cover the ‘automatic’ limitation is given above in the Response to Arguments section and is incorporated by reference. Bigjakkstaffa page 8 teaches that a user incrementally increases and benchmarks, wherein a point is reached “when graphical oddities occur” and “unusual graphical glitches in your games and benchmarks” occur. “For example, textures may flicker a lot of colored dots may

appear." Therefore, this teaching clearly suggests monitoring 'pixel errors', wherein the term is very broad, and clearly would include such things as 'flicker', the appearance of 'colored dots', and the like)

-Automatically determining a safe set of overclocking parameters within said set of supported overclocking parameters passing said stress tests wherein said set of overclocking parameters passes said stress test if the number of pixel errors is below a threshold level. (Bigjakkstaffa teaches that a user incrementally increases and benchmarks, wherein a point is reached "when graphical oddities occur" and "unusual graphical glitches in your games and benchmarks" occur. "For example, textures may flicker a lot of colored dots may appear." These clearly teach that pixel errors occur and/or become worse when the system exceeds a safe, optimal, or desirable level, wherein Bigjakkstaffa states (page 8) that when such a point is reached, "My friends, you have hit the ceiling for your overclock. All that you can do now is to lower your clock speed by 15MHz or so to a point you know to be stable from your incremental method." Clearly, this is a teaching of incrementally increasing the various parameters until the number or quantity of visible pixel errors exceeds some point (e.g. becomes visually noticeable). This teaches the idea of a threshold. Bigjakkstaffa describes counting pixel errors and incrementing said clock rate comprises incrementing said clock rate until a number of pixel errors exceeds a pre-selected number of pixel errors (see Bigjakkstaffa, page 8, last paragraph, wherein the clock rate is incrementally increased to form a new set of overclocking parameters, and a stress test in the form of games and benchmarks are performed at the new clock rate, and further wherein when

the test exceeds a preselected number of errors, in this case at least one such as a graphical glitch or a showing of colored dots/pixels, then the clock rate is no longer incremented),

Bigjakkstaffa describes executing a test program sequence of graphical operations; and determining a number of pixel errors generated by a graphics pipeline of said graphics system; wherein said set of overclocking parameters passes said stress test if said number of pixel errors is no greater than a preselected number of pixel errors (see Bigjakkstaffa, page 8, last paragraph, wherein the clock rate is incrementally increased to form a new set of overclocking parameters, and a stress test in the form of games and benchmarks are performed at the new clock rate, and further wherein when the test exceeds a preselected number of errors, in this case at least one such as a graphical glitch or a showing of colored dots/pixels, then the clock rate is no longer incremented).

Bigjakkstaffa teaches most of the limitations of the instant claim. However, Bigjakkstaffa fails to explicitly teach the use of automatic testing of sets of parameters and then determining if a set of overclocking parameters passes said stress test if the number of pixel errors is below a threshold level. Bigjakkstaffa does teach determining the number of texture tears and/or visual artifacts as measurements of whether or not the overclocking is successful.

One of ordinary skill in the art would therefore turn to references within the arts of memory and processor overclocking to determine how such overclocking could be improved.

Feierbach clearly teaches the use of automatic testing for parameter sets with respect to memory speed, and checking to see if a set of overclocking parameters passes said stress test.

Feierbach clearly teaches automatically checking a set of parameters with respect to memory clock rates and evaluating errors up to a threshold.

In the memory arts, Feierbach teaches the use of a binary test pattern being written to a DRAM. Initial or default values are chosen for the variable(s) that are not optimized, but one variable is set to an aggressive (e.g. high) value (e.g. step 50, Figure 2) [0007,0013]. If the test pattern is read from the DRAM and there are no errors, e.g. it matches the original binary test pattern [0015] (steps 51-53), then the system proceeds to determine if the opposite test pattern can be held (steps 54-56)[0016-0018]. If so, then the system scales the value to a higher one (step 59)(if the loop is on the first pass). If at any point an error is detected, the system goes into a loop check mode, where the aggressively scaled value is decreased (see path step 66). However, if the resolution is less than a predetermined resolution threshold from the calculated upper refresh period limit, then the system passes that stable value on to the registers for use [0018-0024], the procedure is illustrated in Figures 3(a)-(e). This procedure is then used to determine optimal values for the other parameters [0028] - that is, scaling the values up from the initial to an optimal level, and/or scaling them down to an optimal level. These techniques are applicable to DRAM clock rates, in that the signals being adjusted are generated by control circuits that have system clocks therein [0006].

Therefore, adjusting the system clocks is clearly in keeping with the spirit of the invention [0029].

That is, Feierbach teaches an automated method of scaling memory clock rates (e.g. CAS / RAS are created from system clock rates) and testing them to find the optimal values, and tests sets of parameters to find the optimal set.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Bigjakkstaffa to use the techniques of Feierbach to utilize automatic testing of parameter sets for memory clock rate to achieve the lowest power consumption (e.g. longer intervals between refreshes) and more efficient use of memory.

Bigjakkstaffa and Feierbach fail to expressly teach automatically testing sets of parameters that involve processor speed and memory speed together and then determining whether or not they exceed a set threshold [it is noted that Bigjakkstaffa teaches that one standard for evaluating processor and memory speeds is thermal control (e.g. when the processor is too hot, thermal damage may occur (see Bigjakkstaffa page 8 and the like)).], and also fail to expressly teach automatically running graphical benchmarks (e.g. the system of Bigjakkstaffa is manually actuated).

Cooper clearly teaches automatically stress testing a processor over different sets of memory and processor parameters. Specifically, Cooper teaches that a system has a maximum safe operating temperature [0003] ('junction temperature specification')[0029](wherein if the temperature exceeds said safe limit, damage will

occur to the chip). Cooper teaches that thermal sensors can be on-die [0031](e.g. junction temperature monitors)('maximal allowable die temperatures'). The system of Cooper (Figure 3, stress tests – including graphics controller to graphics controller or graphics controller to local memory – [0035-0036]) very clearly tests different clock-frequency / bandwidth (e.g. bus speed) combinations (Abstract).

Cooper also teaches the use of graphics stress software to determine maximum bandwidths and speed (fill rates, etc), wherein such stress tests and the instant embodiments can be applied upon graphics controllers and the like (e.g. GPUs), wherein this clearly must require automatically running benchmarks to obtain information such as maximum fill rate and the like. Figure 6 also shows such stress tests.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Bigjakkstaffa and Feierbach in light of the teachings of Cooper because they allow easier, more direct visualization of test results, temperature versus speed and bandwidth plots, and the like, wherein such can be displayed as a GUI [0027,0033,0040], wherein this use of a GUI and the system of Cooper increases efficiency and requires minimal customer input or direction, and the like [0040], again with the integral sensors being the most effective [0039].

Clearly, monitoring graphics parameters (such as fill rate) are suggested by Cooper, and they are done on an automated basis, thusly supporting the holding above under *Venner* that one of ordinary skill in the art would have used an automated means to observe such parameters for benchmarked test sequences.

With regard to claim 3, Bigjakkstaffa describes adjusting a graphics processor core clock rate (see Bigjakkstaffa, pages 5 and 6, wherein the overclocking parameters being adjusted by the user are the clock of the GPU core and the clock of the GPU memory).

With regard to claim 4, Bigjakkstaffa describes adjusting a graphics memory clock rate (see Bigjakkstaffa, pages 5 and 6, wherein the graphics memory clock rate adjusted is the clock of the GPU memory, or graphics memory).

With regard to claim 6, incrementing a clock rate of said graphics system to form new sets of overclocking parameters; and applying said stress test for each incremental increase in said clock rate; said clock rate incremented until a number of errors associated with said stress test exceeds a preselected number of errors (see Bigjakkstaffa, page 8, last paragraph), wherein the clock rate is incrementally increased to form a new set of overclocking parameters, and a stress test in the form of games and benchmarks are performed at the new clock rate, and further wherein when the test exceeds a preselected number of errors, in this case at least one such as a graphical glitch or a showing of colored dots, then the clock rate is no longer incremented). As stated above, Cooper and Feierbach teach automatically performing such tests, and the rejection to claim 1 is incorporated in its entirety.

With regard to claim 10, Bigjakkstaffa describes selecting a maximum safe clock rate of a graphics processing unit (see Bigjakkstaffa, page 8, last paragraph, wherein the clock rate of a GPU is incrementally increased to form a new set of overclocking parameters, and a stress test in the form of games and benchmarks are performed at

the new clock rate, and further wherein when the test exceeds a preselected number of errors, in this case at least one such as a graphical glitch or a showing of colored dots, then the clock rate is no longer incremented, instead, the clock rate is decremented and selected by the user to the last stable clock rate which is the maximum safe GPU clock rate). Further, examiner submits that a maximum safe operating frequency would constitute the lesser of: unacceptable pixel error threshold reached, or goal temperature reached or exceeded (based on the teachings of Cooper).

With regard to claim 11, Bigjakkstaffa describes selecting a maximum safe clock rate of a graphics memory (see Bigjakkstaffa, page 8, last paragraph, wherein the clock rate of a graphics memory is incrementally increased to form a new set of overclocking parameters, and a stress test in the form of games and benchmarks are performed at the new clock rate, and further wherein when the test exceeds a preselected number of errors, in this case at least one such as a graphical glitch or a showing of colored dots, then the clock rate is no longer incremented, instead, the clock rate is decremented and selected by the user to the last stable clock rate which is the maximum safe memory clock rate). Further, examiner submits that a maximum safe operating frequency would constitute when unacceptable pixel error threshold was reached (e.g. test sequence corrupted (Feierbach)), which is clearly taught to be done in an automated fashion by the references.

With regard to claim 12,

The other limitations above are similar to those found in the rejection to claim 1, which is incorporated by reference. The only significant differences are discussed below:

-The determining a maximum clock rate for each of said at least one clock for which said graphics system has a number of errors below a threshold level; and setting said at least one clock rate at said maximum clock rate(s) (see Bigjakkstaffa, page 8, last paragraph, wherein the clock rate is incrementally increased to form a new set of overclocking parameters, and a stress test in the form of games and benchmarks are performed at the new clock rate, and further wherein when the test exceeds a preselected number of errors, in this case at least one such as a graphical glitch or a showing of colored dots, then the clock rate is no longer incremented, instead, the clock rate is decremented to the last stable clock rate which is the maximum clock rate(s)).

Specifically, the Feierbach reference teaches testing of each the three parameters separately (RAS, CAS, and refresh rates) and finding the maximum safe or stable values for each reference separately. Therefore, in light of the teachings of Feierbach, it would have been obvious to optimize each variable (clock rate) and to set them all to their maximum safe values (as per Feierbach). Feierbach further teaches setting all of the variables (RAS, CAS, refresh rate) to their maximum safe values.

The other limitations above are similar to those found in the rejection to claim 1, which is incorporated by reference, and the motivation and rationale is taken therein.

With regard to claim 13, Bigjakkstaffa describes receiving an input from a control panel of a graphical user interface (see Bigjakkstaffa, page 6, wherein the system

tweaks control panel, specifically the overclocking tab, receives user input when the user drags the core clock and memory clock sliders).

With regard to claim 14, Bigjakkstaffa describes displaying said graphical user with updated overclocking parameters (see Bigjakkstaffa, page 8, last paragraph, and page 9, wherein the updated overclocking parameters of page 8 are shown by the overclocking control panel on page 9).

The rationale for modifying Bigiakkstaffa to cover the 'automatic' limitation is given above in the Response to Arguments section and is incorporated by reference.

With regard to claim 15, Bigjakkstaffa describes incrementing a core clock rate of a graphics processing unit rate (see Bigjakkstaffa, page 6, which describes incrementing a core clock rate of a graphics processing unit by dragging the core clock slider up in 10Mhz increments).

With regard to claim 16, Bigjakkstaffa describes incrementing a memory clock rate of a graphics memory (see Bigjakkstaffa, page 6, which describes incrementing a memory clock rate of a graphics processing unit by dragging the memory clock slider up in 10Mhz increments).

With regard to claim 17, Bigjakkstaffa describes incrementing a first clock rate by a first preselected increase in clock rate (see Bigjakkstaffa, page 6, which describes incrementing a core clock rate of a graphics processing unit by dragging the core clock slider up in preselected 10Mhz increments); and incrementing a second clock rate by a second preselected increase in clock rate (see Bigjakkstaffa, page 6, which describes

incrementing a memory clock rate of a graphics processing unit by dragging the memory clock slider up in preselected 10Mhz increments).

With respect to claim 20, Bigjakkstaffa fails to teach this limitation, where Cooper clearly teaches sensing on-chip temperature, and then detecting when the threshold temperature is reached during stress testing, throughput and/or frequency is lowered to maintain said chip temperature at or below said threshold (Figures 4-6, step 402, and the like, Abstract, etc). Motivation taken from the rejection to claim 12 above.

With respect to claim 22, the GeForce4 card taught by Bigjakkstaffa is known to have a GPU with a graphics pipeline (see Gasior). The rejection to claim 12 is incorporated by reference, as this is the system implementing said method. The only difference is that the system claim does not expressly require that the clock(s) all be set at their maximum clock rate(s) and/or parameter(s). In order for the method to execute, a system implementing the method would *prima facie* have modules that did each specific item. Finally, the Cooper reference shows such a module in any case.

With regards to claim 23, the graphics pipeline for each set of overclocking parameters would generate clearly any pixel errors.

With regards to claim 24, clearly the overclocking menu tab on page 3, and see page 5, first paragraph, wherein the user requests overclocking by ticking the checkbox marked "Enable driver level hardware overclocking" near the top of the window of the overclocking control panel) and wherein said user inputs said request to said control panel (see Bigjakkstaffa, Feierbach, and Cooper, page 6, wherein the user adjusts at least one clock rate to form at least one new clock rate by dragging the core clock and

memory clock sliders up in 10Mhz increments, and wherein the graphical user interface that the user is interacting with is inherently generated by computer executable instructions).

With regards to claim 25, Bigjakkstaffa teaches determining a maximum safe GPU clock rate (e.g. the point at which pixel errors exceed a threshold), as does Cooper (goal temperature).

With regards to claim 26, Bigjakkstaffa teaches determining a maximum safe memory clock rate (e.g. the point at which pixel errors exceed a threshold), as does Cooper (goal temperature)

With regards to claim 27, this merely combines determining the maximum clock rates of the GPU and memory, both of which are shown in Bigjakkstaffa, but not expressly taught. Bigjakkstaffa, Feierbach, and Cooper, page 8, last paragraph, wherein the clock rate of a GPU and the memory rate of the GPU are incrementally increased to form a new set of overclocking parameters, and a stress test in the form of games and benchmarks are performed at the new clock rates, and further wherein when the test exceeds a preselected number of errors, in this case at least one such as a graphical glitch or a showing of colored dots, then the GPU core and memory clock rates are no longer incremented, instead, the memory and GPU clock rates are decremented to the last stable clock rates which are the maximum safe GPU core clock rate and the maximum safe GPU memory clock rate. Wherein the Feierbach reference teaches testing of each the three parameters separately (RAS, CAS, and refresh rates) and finding the maximum safe or stable values for each reference separately.

Therefore, in light of the teachings of Feierbach, it would have been obvious to optimize each variable (clock rate) and to set them all to their maximum safe values (as per Feierbach). Feierbach further teaches setting all of the variables (RAS, CAS, refresh rate) to their maximum safe values.

With regards to claim 28, this is merely the method of claim 12 in computer-readable medium form, and is the system of claim 22 implemented in software form.

With regards to claim 29, the system of Bigjakkstaffa clearly teaches a control panel for making such modifications; the other limitations are the method of claim 12, which is incorporated by reference.

With regards to claim 30, clearly the stated means can be software implemented on a computer, which is the system of Bigjakkstaffa, wherein otherwise this is the method claim of claim 12, as implemented in software in claim 18. Therefore, the software of Bigjakkstaffa as modified in said rejections serves as the recited means or their functional equivalents.

With regards to claim 31, this is the method of claim 1, implemented in software on a computer. Clearly the method of claim 1 is embodied in software on a computer, wherein Bigjakkstaffa teaches the other limitations, wherein the recited means are the software of Bigjakkstaffa and/or their functional equivalents.

With regards to claim 32, clearly Bigjakkstaffa and Cooper teach displaying control panel means with overclocking parameters, but fails to expressly teach displaying maximum safe overclocking parameters.

Bigjakkstaffa describes control panel means for displaying maximum safe overclocking parameters (see Bigjakkstaffa, pages 6 and 7, wherein the user clicks on the “test” button, and wherein further tests are performed by running a 3dmark bench, and if no visual artifacts or texture tearing appears, then the user defined GPU core and GPU clock speeds pass the test and are determined to be a safe set of overclocking parameters, and said safe overclocking parameters are displayed on the overclocking tab of the system tweaks control panel at page 9 of Bigjakkstaffa). Additionally it would have been obvious to one of ordinary skill in the art at the time the invention was made to display such parameters to the user so that the operating profile could be easily understood.

Claims 2 and 21 are rejected under 35 USC 103(a) as unpatentable over Bigjakkstaffa, Feierbach, and Cooper as applied to claim 1 and further in view of Culbert et al (US PGPub 2005/0049729 A1).

As to claim 2, Bigjakkstaffa clearly suggests adding a fan to the GPU to ensure effective cooling, but does not teach changing the fan speed.

Culbert clearly teaches such limitations in [0011-0013], in that when the processor (inclusive of GPU) speed changes, the fan speed will change as well, as in [0041], where the CPU voltage and frequency control are provided to have many states, wherein the cooling fans have speed control as well, as dependent upon processor speed and the temperature as required [0069, 0096]. The thermal manager keeps the

components under a desired temperature by controlling fan speed, and the like. It would have been obvious to one of ordinary skill in the art to modify Bigjakkstaffa, Feierbach, and Cooper to vary fan speed with increases in processor speed and decrease it with decreases therein because the heat produced will be greater or lesser as the case may be above because such variance increases cooling and helps keep the processing unit(s) under the goal (maximum) temperature, thusly allowing better load testing a la Cooper.

As to claim 21, Bigjakkstaffa, Feierbach, and Cooper do not expressly teach this limitation. Culbert clearly teaches that the processor core clock rate has a set voltage associated with it, e.g. a higher operating frequency requires a higher voltage because the power consumption increases, particularly the dynamic version. Therefore, changing states would result in a changed voltage for each frequency as noted above (Abstract, [0011-0013]). Motivation and rationale taken from the rejection to claim 2 above.

Claim 5 is rejected under 35 USC 103(a) as unpatentable over Bigjakkstaffa, Feierbach, and Cooper as applied to claim 1 and further in view of Culbert and Kao (US 6,622,254).

As to claim 5, Bigjakkstaffa, Feierbach, and Cooper collectively teach adjusting memory timings, wherein the CAS / RAS rates of DRAM are derived from system clocks found within the memory module (see Cooper as above), and adjusting the memory

clock frequency would obviously change the system clocks generating aforementioned, thusly changing the memory timings. They further teach adjusting at least one clock rate to form at least one new clock rate.

They fail to expressly teach setting a chip voltage, memory timing, and fan speed for each said at least one new clock rate.

Culbert very clearly teaches that the processor has various states that include both chip voltages and frequencies (e.g. state 1 has frequency 1 and voltage 1). Culbert, as in the rejection to claim 2 above, clearly teaches varying fan speeds based on processor operating conditions.

Therefore, as noted above, a given processor operating frequency will have a linked operating voltage (e.g. higher operating frequency requires more power) as per the teachings of Culbert. Concomitant with higher power usage, the thermal load of the GPU or CPU will increase, thusly necessitating a higher fan speed associated with a state having higher operating frequency. Motivation and rationale are taken from the rejection to claim 2 above.

Culbert fails to teach that adjusting the frequency of the processor requires new memory timings.

Kao clearly teaches in the Background (1:20-30) that: "The external frequency is the speed at which the cache and the main memory communicate with the CPU. Changing the external frequency means to change the bus speed. Increase the external frequency one step at a time is the most successful way to overclock a CPU."

Later on Kao teaches that one of the steps of overclocking a CPU may involve (step 8, 2:8-9) "Try some other memory timings in the BIOS setup, if necessary".

Therefore, Kao clearly teaches that changing the external frequency is the most effective manner to overclock the CPU, and if the operating speed of the FSB is changed, clearly that changes the manner in which it interacts with the memory, which **may** therefore involve changing memory timings. In light of the teachings of Kao, it would be obvious that if the processor (and FSB) frequency increased, then memory timings should be re-optimized. Motivation and rationale is taken from the nature of the reference and the problem to be solved, in that the reference directly addresses the **best** or most successful manner in which to perform the task. Therefore, under the preponderance of the evidence and reasonable examiner tests, as well as under the standards set forth by *In re Fine* and *In re Cartwright*, examiner has met the burden of showing a *prima facie* case against the instant claims.

Claims 8 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bigjakkstaffa, Feierbach, and Cooper as applied to claim 1, in view of Catenary Systems.

With regard to claim 8, Bigjakkstaffa, Feierbach, and Cooper are relied upon for describing all of the limitations of parent claim 1, as discussed in the 103(a) rejection above. Bigjakkstaffa, Feierbach, and Cooper fail to explicitly describe writing to a three dimensional surface; performing an exclusive or operation; and determining uniformity,

as recited in claim 8. However, official notice is hereby taken that writing to a three dimensional surface is notoriously well known in the art, and Catenary teaches performing an exclusive or operation and determining uniformity (see Catenary, page 1, wherein an XOR operation is performed on two images to determine their similarities or uniformity and differences).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify Bigjakkstaffa, Feierbach, and Cooper to incorporate the XOR operations and uniformity determination of Catenary Systems, because XORing two images generated by the graphics stress test of Bigjakkstaffa, Feierbach, and Cooper allows one to determine when the frequency of the GPU is set too high, as, for example, when two 3D generated test images are determined to be non-uniform by a predetermined percentage. The rationale for modifying Bigiakkstaffa to cover the 'automatic' limitation is given above in the Response to Arguments section and is incorporated by reference.

With regard to claim 19, Bigjakkstaffa, Feierbach, and Cooper are relied upon for describing all of the limitations of parent claim 12, as discussed in the 103(a) rejection above. Bigjakkstaffa, Feierbach, and Cooper fail to explicitly describe writing to a three dimensional surface; performing an exclusive or operation; and determining uniformity, as recited in claim 19. However, official notice is hereby taken that writing to a three dimensional surface is notoriously well known in the art, and Catenary teaches performing an exclusive or operation and determining uniformity (see Catenary, page 1,

wherein an XOR operation is performed on two images to determine their similarities or uniformity and differences).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify Bigjakkstaffa, Feierbach, and Cooper to incorporate the XOR operations and uniformity determination of Catenary Systems, because XORing two images generated by the graphics stress test of Bigjakkstaffa, Feierbach, and Cooper allows one to determine when the frequency of the GPU is set too high, as, for example, when two 3D generated test images are determined to be non-uniform by a predetermined percentage. The rationale for modifying Bigiakkstaffa to cover the 'automatic' limitation is given above in the Response to Arguments section and is incorporated by reference.

Claims 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bigjakkstaffa, Feierbach, Cooper, and Gasior ("TR's GeForce Ti 4200 comparo."). As stated in the previous Office Action, Gasior inherently is combinable with Bigiakkstaffa as below (e.g. Gasior teaches that the graphics card tested by Bigjakkstaffa has a graphics pipeline and other details of the hardware).

With regard to claim 18, Bigjakkstaffa, Feierbach, and Cooper are relied upon for describing all of the limitations of parent claim 17, as discussed in the 103(a) rejection above.

The references above do not explicitly teach counting pixel **bit** errors. However, textures and color dots do comprise pixels, and Bigjakkstaffa clearly teaches counting pixel errors. Now, applicant does not expressly define how a “pixel bit error” is different than a pixel error. Therefore, no applicant-specific definitions are found to be controlling in this situation, even in view of *Phillips v. AWH*. Next, the claims must be given the broadest reasonable interpretation with respect to the specification (*In re Morris*), which is necessarily broader and different than that given by courts. It has been held repeatedly by the CAFC that applicant’s claims are not bound to the only disclosed embodiment in the specification but also **functional equivalents** and the like, even if means-plus-function language is not expressly included.

The added references – Cooper particularly – clearly teaches testing a system such that it can be overclocked up to the maximum safe threshold, as does Feierbach. Feierbach continues to change parameters until the maximum safe threshold can be determined for overclocking purposes. That being said, Cooper and Feierbach at least suggests the idea of using a threshold. Now, in the case of Bigjakkstaffa, a human being determines the quantity of visual errors and determines what an acceptable level of errors are. Even in light of applicant’s Remarks on page 10, regardless of how subjective such a determination may be, the human being or user does in fact make such a determination. Now, the concept of using an objective threshold for such determinations is obvious in light of Cooper and Feierbach. Further, in light of *In re Vanner*, the computer could obviously do the thresholding and judgment automatically,

and it would make such judgments. Applicant's remarks overlook the fact that the human being in fact performs two roles.

Now, a pixel **bit** error can be quantified in many different ways. However, applicant's specification would seem to indicate that a pixel bit error could broadly be construed as an error in the pixel (e.g. a mis-rendered pixel with some indication (visually) that it was not correct). A pixel **bit** error would result in an incorrectly rendered pixel all the same (e.g. a pixel is standardly represented as 32 bit RGBA (8 bit red, 8 bit green, 8 bit blue, 8 bit alpha). Any error in those would result in the pixel being the wrong color or transparency, which would inherently constitute a pixel error. Indeed, a 'torn texture' or otherwise would still result in a **visually noticeable** defect of some kind –even if such a defect were only visible to a user with very high visual acuity without color-blindness. Such details – the amount of visual acuity and/or ability to see color – are quite beside the point in the interpretation of the Bigjakkstaffa reference, because the obviousness determination is not made with respect to whether or not a human user has some sort of disability or enhancement that would (not) allow such a user to detect visual errors – it is merely whether or not the reference **suggests** such a limitation). Examiner believes those arguments to be misdirected for at least the above reasons.

Now, to the specific point of pixel bit errors, Bigjakkstaffa, Feierbach, and Cooper describes counting pixel bit errors (see Bigjakkstaffa, Feierbach, and Cooper, page 8, last paragraph, wherein the clock rate is incrementally increased to form a new set of overclocking parameters, and a stress test in the form of games and benchmarks are performed at the new clock rate, and further wherein when the test exceeds a

preselected number of errors, in this case at least one such as a graphical glitch or a showing of colored dots/pixels, then the clock rate is no longer incremented), but fails to explicitly describe a graphics pipeline, as further recited in claim 18. However, Gasior teaches that the GeForce4 Ti 4200 graphics card tested by Bigjakkstaffa, Feierbach, and Cooper inherently has a graphics pipeline (see Gasior, page 1, fourth paragraph). The rationale for modifying Bigiakkstaffa to cover the 'automatic' limitation is given above in the Response to Arguments section and is incorporated by reference.

### ***Conclusion***

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Eric Woods whose telephone number is 571-272-7775. The examiner can normally be reached on M-F 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ulka Chauhan can be reached on 571-272-7782. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Eric Woods

November 16, 2006



KEE M. TUNG  
SUPERVISORY PATENT EXAMINER